Implementation of the IEEE 1588 Precision Time Protocol for Clock Synchronization in the Radio Detection of Ultra-High Energy Neutrinos

Undergraduate Research Thesis

Presented in partial fulfillment of the requirements
for graduation with *research distinction* in Physics in the undergraduate
colleges of The Ohio State University

by

Spoorthi Nagasamudram

The Ohio State University

April 2018

Project Advisor: Professor Amy Connolly, Department of Physics

Abstract

Spoorthi Nagasamudram

Implementation of the IEEE 1588 Precision Time Protocol for Clock Synchronization in the Radio Detection of Ultra-High Energy Neutrinos

In this project, we have implemented a high precision time synchronization protocol called precision time protocol in one of the power distribution boards for the Askaryan Radio Array. We discuss the preliminary stages of execution of the protocol and some results. This type of time synchronization will give us synchronized clocks between stations. We can achieve precision on the order of nanoseconds which is suitable for our experiments. The advantage of high precision time synchronization is that it will allow us to conduct more efficient time coincidence searches for neutrino events between stations. It can also potentially lead to searches between the IceCube observatory and the Askaryan Radio Array.

Acknowledgements

I would like to thank Prof. Amy Connolly for everything. Without her, I would not be where I am today. I would like to extend a special thanks to my mentor Dr. Patrick Allison who spent hours helping me. I want to also thank Prof. Jordan Hanson and Prof. Carl Pfendner for helping me build the raytracer. I would like to thank Prof. James Beatty for serving on the thesis committee. Finally, I would like to thank Brian Clark, Oindree Banerjee and Sam Stafford for the wonderful support...

Contents

List of Figures

1 Ultra High Energy (UHE) Neutrinos

1.1 Neutrino Astronomy

Ultra-high energy (UHE) neutrinos are a topic of great interest in the field of particle astrophysics. They have extremely high energies ranging from $10^{18} - 10^{21}$ eV. They are able to travel long distances from their sources because they do not interact with electromagnetic fields such as galactic magnetic fields and other particles like cosmic rays since neutrinos are neutrally charged. Therefore, detecting UHE neutrinos can help humans probe into the astrophysical sources such as supernovae that produce such energetic neutrinos.

1.2 Detection of UHE Neutrinos

1.2.1 The Askaryan Effect

When an energetic particle interacts with a dielectric medium, it produces a shower of charged particles moving at very high speeds. When the secondary charged particles travel faster than the speed of light in the dielectric medium, electromagnetic radiation is produced. The radiation is coherent in wavelengths longer than the Moliére radius of the medium. This effect is called the Askaryan Effect, named after the scientist who first theorized it, Gurgen Askaryan, and is one of the most cost-effective ways to detect UHE neutrinos.

1.2.2 Motivation for Using Ice as the Dielectric Medium

UHE neutrinos have an extremely low flux ($1\ neutrino/km^2/century$) and interact very rarely. In order to compensate for the low flux, a very large detector volume (thousands of cubic kilometers) is required to detect them. It would be very expensive to build such a large detector in a laboratory. The large volume of naturally-occurring ice in Antarctica is not only cheaper to use but also more convenient. The Moliére radius of ice is about 10 cm, placing the Askaryan radiation in ice in the radio regime. The attenuation length of radio in ice is about 1 km which makes it possible to detect radio signals on such large detector scales. Therefore, ice is a suitable dielectric medium to detect UHE neutrinos.

1.2.3 The Askaryan Radio Array (ARA)

The Askaryan Radio Array (ARA) is a radio detector of UHE neutrinos and is located at the South Pole. It consists of an array of antennas located deep in the ice at a depth of 200 m. With five stations installed and running, ARA is currently able to look for radio signals from UHE neutrinos in an area covering approximately 5-6 square kilometers. The goal is to have 37 stations installed covering upto 100 square kilometers. ARA is in close proximity to the famous IceCube observatory. IceCube has previously detected UHE neutrinos upto a few 10^{15} eV that were likely produced from outside our solar system.

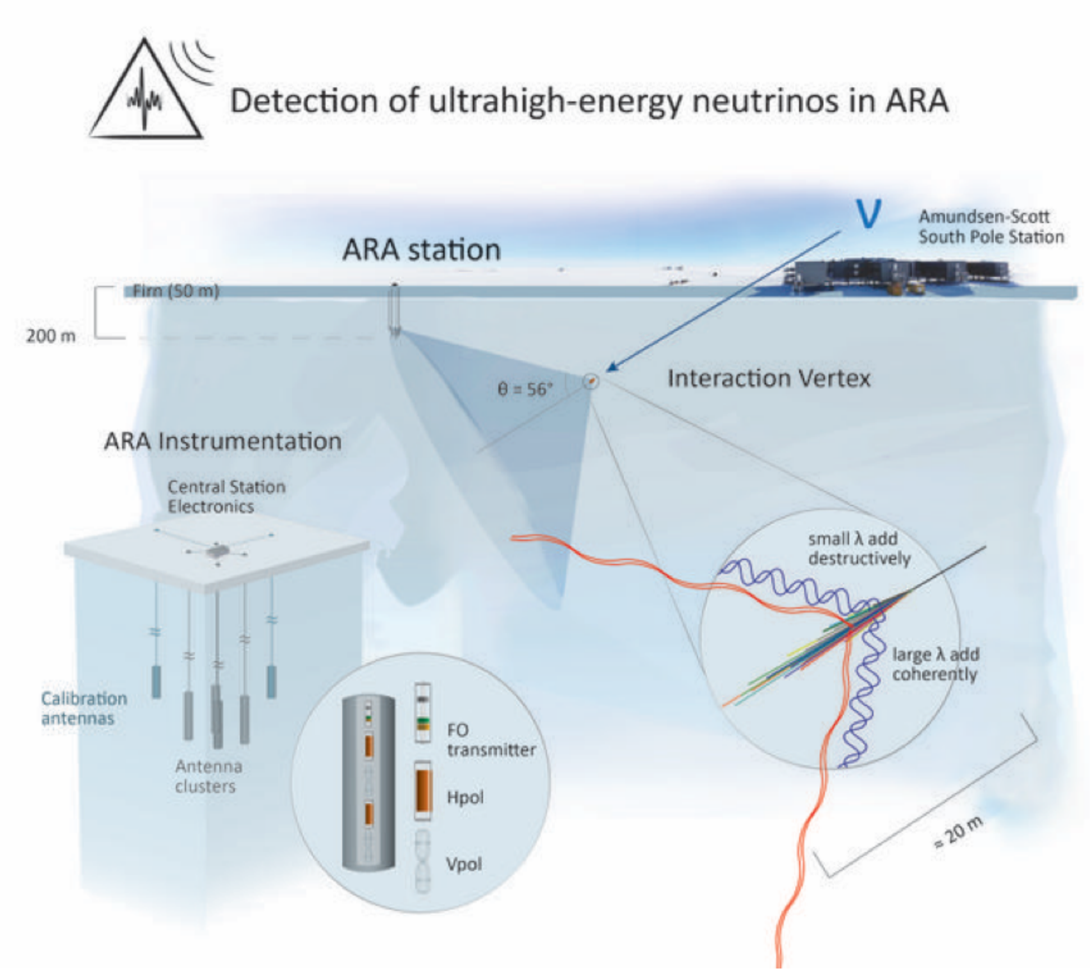

FIGURE 1.1: The Askaryan Radio Array detecting askaryan radiation in ice

2 Precision Time Protocol

As is the case with any high-energy physics experiment, there is a lot of background noise in the events that ARA detects. Most of the noise is thermal noise which comes from below the ice surface. Timing synchronization between the ARA stations is helpful in eliminating the background which will allow us to lower the energy threshold when searching for neutrinos. Timing synchronization between IceCube and ARA is also possible and can help eliminate background events pertaining to conditions unique to ARA. Precision Time Protocol (PTP) is an IEEE 1588 standard that can provide timing synchronization in the range of nanoseconds. Due to its high accuracy, it is suitable to use PTP to provide timing synchronization.

2.1 Method to Synchronize Clocks Using PTP

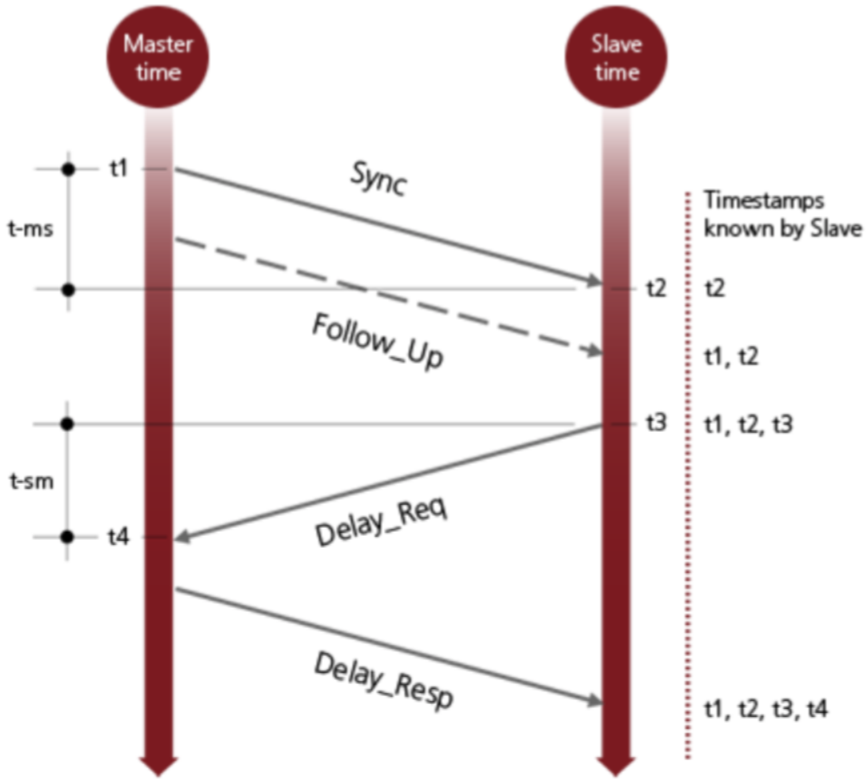

FIGURE 2.1: A simplified view of precision time protocol (http://www.chronos.co.uk/files/pdfs/cal/TechnicalBrief-IEEE1588v2PTP.pdf)

$t1$ =Master time when sync message was sent.

$t2$ =Slave time when sync message was received.

$t3$ =Slave time when delay request message was sent.

$t4$ =Master time when delay request message was received.

Let us consider trying to synchronize the clocks by exchanging a series of synchronizing messages. The sync message is a synchronization message sent by the master clock to the slave clock. The slave notes down the slave time when it receives the sync message ($t2$). The follow-up message is used to accurately convey the master time when the sync message was sent ($t1$). The master-to-slave propagation delay is given by:

$$t_{ms} = t2 - t1$$

The delay request message is a synchronization message sent by the slave to the master. It contains the slave time when the delay request was sent to the master ($t3$). The delay response message is a synchronization message sent by the master to the slave upon receiving a delay request message. It contains the master time when the delay request was received by the master ($t4$). The slave-to-master propagation delay is given by:

$$t_{sm} = t4 - t3$$

The mean propagation delay is calculated to be:

$$t = \frac{(t2 - t1) + (t4 - t3)}{2}$$

Therefore, the clock offset can be calculated as:

$$t2 - t1 - t$$

2.2 PTP Message Format

A basic template for a PTP message is shown below. The PTP header is common to all PTP messages. However, the content of the body may differ depending upon the type of message.

FIGURE 2.2: PTP Message Format

PTP Header

The PTP Header consists of 34 bytes. The format of the header is shown in Fig. 2.3.

PTP Message Header Format									
Bits								Octets	Offset
7	6	5	4	3	2	1	0		
transportSpecific				messageType				1	0
Reserved				versionPTP				1	1
messageLength								2	2
domainNumber								1	4
Reserved								1	5
Flags								2	6
correctionField								8	8
Reserved								4	16
sourcePortIdentity								10	20
sequenceID								2	30
controlField								1	32
logMessageInterval								1	33

FIGURE 2.3: PTP Header Format

Message Type – refers to the type of PTP message (e.g. sync, follow-up, delay-request).

Message Length–refers to the length of the PTP message in bytes.

Domain Number–refers to the group of clocks being synchronized with each other. Clocks with different domain numbers are not necessarily synchronized with each other.

Source Port Identity–contains information about the clock ID and the port that the message is originating form.

Sequence ID–refers to the sequence number of each type of message.

Control–is another way to find out about the type of PTP message. Different PTP messages have different control values.

2.3 Different Types of PTP Messages

Announce Message

The announce message is used to indicate to all clocks about the master clock. This allows for the master-slave hierarchy to be established. The format of the announce message is shown in Fig. 2.4.

Announce Message Format									
Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header (13.3)								34	0
originTimestamp								10	34
currentUtcOffset								2	44
Reserved								1	46
grandmasterPriority1								1	47
grandmasterClockQuality								4	48
grandmasterPriority2								1	52
grandmasterIdentity								8	53
stepsRemoved								2	61
timeSource								1	63

FIGURE 2.4: Announce Message Format

Sync Message

The sync message is sent by the master to the slave clock. PTP messages can be timestamped in a one-step or a two-step process. In the two-step process, there is a second message (follow-up message) that is sent after the sync message. The follow-up message contains the actual master time when the sync message was sent. In the one-step process, there is no second follow-up message and the master time when the sync message was sent in contained in the sync message itself.

Sync Message Format									
Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header (13.3)								34	0
originTimestamp								10	34

FIGURE 2.5: Sync Message Format

Follow-Up Message

The follow-up message is sent if the master clock is a two-step clock. It contains the actual master time when the sync message was sent.

Delay Request Message

The delay request is sent by the slave clock to the master clock. It contains the slave time when the message was sent.

Follow_Up Message Format									
Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header (13.3)								34	0
preciseOriginTimestamp								10	34

FIGURE 2.6: Follow-Up Message Format

Delay_Req Message Format									
Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header (13.3)								34	0
originTimestamp								10	34

FIGURE 2.7: Delay Request Message Format

Delay Response Message

The delay response is sent by the master clock to the slave clock. It contains the master time when the delay request was received.

Delay_Resp Message Format									
Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header (13.3)								34	0
receiveTimestamp								10	34
requestingPortIdentity								10	44

FIGURE 2.8: Delay Response Message Format

2.4 PTP Timestamps

PTP follows the epoch time with UTC reference. Since PTP allows for nanosecond scale precision, the timestamp shows the seconds as well as nanoseconds. The second field in the timestamp is usually a 48-bit integer whereas the nanosecond field in the timestamp is a 32-bit integer.

3 Design and Implementation of PTP Within the Framework of ARA

The main goal of this research project was to implement PTP within the framework of ARA. We used one of the power distribution boards for ARA to do this. The master clock was set up locally on caesar.mps.ohio-state.edu and the microcontroller in the power distribution board served as the slave clock. Essentially three main steps were executed to implement PTP in this setup. In the first step, we programmed the microcontroller to send and receive PTP messages. In the second step, we extracted the timestamps from both the master and slave clocks. In the third and final step, we designed a program to calculate the offset between the master clock and the slave clock.

3.1 Sending and Receiving PTP Messages Through the ARA Power Distribution Board

The PTP packets are sent over UDP over IPv6 over Ethernet. The incoming messages are parsed. We can also write and send PTP messages to a desired IP address. It is possible to monitor PTP messages using PTP Trackhound or Wireshark (PTP Trackhound:https://www.ptptrackhound.com, Wireshark:https://www.wireshark.org). An example of a PTP packet shown in PTP Track Hound is shown below.

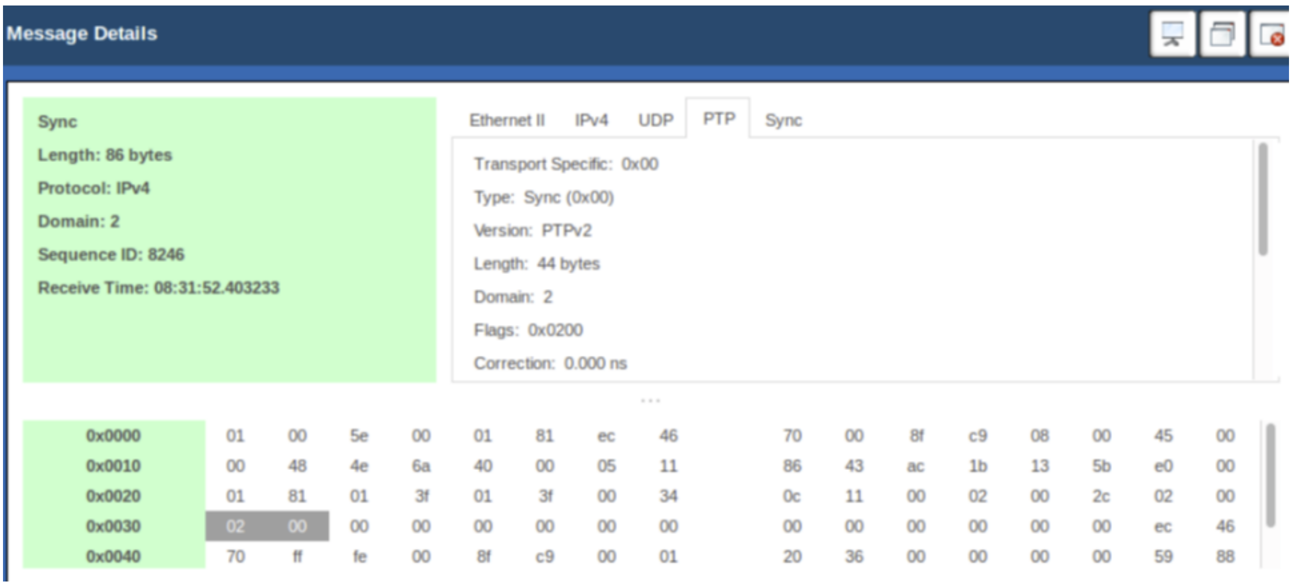

FIGURE 3.1: A PTP Trackhound window showing a sync message

The advantage of using PTP Trackhound/Wireshark is that it shows all the bytes (hexadecimal format) in the packet. This is especially useful in the case of debugging.

3.2 Getting Timestamps

PTP timestamps are often stored in the incoming PTP messages and can be gotten by parsing them. PTP timestamps contain a seconds field and a nanoseconds field. They contain the time (with respect to a particular clock) at which a message was sent or received. In order to calculate the clock offset, we need four timestamps:

Master time when sync message was sent

This is the time with respect to the master clock when the sync message was sent. The master clock that we used was a two-step clock. As explained in section 2.3, the timestamp of the sync message can be found in the last ten bytes of the follow-up message.

Slave time when sync message was received

This is the time with respect to the slave clock when the sync message was received by it. The timestamping happens in the hardware. The timestamp is handed over along with the incoming packet when it is parsed.

Slave time when delay request message was sent

This timestamp represents the time with respect to the slave clock when the delay request message is sent. The timestamping also occurs in the hardware. However,the mechanism for timestamping outgoing packets is slightly different than for incoming packets. It should be said here that we have ignored the first few timestamps in this process. This is because of the fact that there is a variable called TivaTxTimestampDone in the code that needs to be true for the microcontroller to timestamp outgoing packets. This variable is set to false in the initial few packets.

Master time when delay response message was received

This timestamp represents the time with respect to the master clock when the delay response message was received by it. The last ten bytes of the delay response message contain the delay response timestamp.

3.3 Calculating Offset Between Master and Slave Clocks

There are two phases in calculating the offset between the master and slave clocks.

Phase 1: the master-slave hierarchy must be established using the announce message.

Phase 2: After the master-slave hierarchy has been established, the exchange of PTP timing messages can start. This phase consists of two steps:

1. Calculating the mean propagation delay between the two clocks. This is done using the delay request-response mechanism which uses the sync, follow-up, delay request and delay response messages to calculate the mean propagation delay.

2. Calculating the offset using the sync message, follow-up message and the mean propagation delay.

Phase 1

The statistics of the master clock is provided to all the slave clocks by the announce message. All incoming PTP messages are parsed and read into a buffer. We can then identify what type of a message it is (announce, sync, follow-up, etc) by looking at the header.

Phase 2

Step 1: Calculating the mean propagation delay

After the master-slave hierarchy has been established, we are now ready to exchange a series of PTP timing messages to calculate the offset between the master and slave clocks.

A sync message is sent by the master clock to the slave clock. It contains the master time when the sync message was sent. If the master is a two-step clock, this timestamp is sent in a second follow-up message. Following the method outlined in section 3.2, we can extract the master time when the sync message was sent as well as the slave time when the sync message was received by the slave clock.

Next, a delay request packet is written and sent to the master clock. It should be said here that the format of the sent message should match that of the delay request message shown in fig.2.7 for the master to recognize it as a PTP delay request message. The delay request message contains the slave time when the message was sent. We can extract this time by following the steps in section 3.2.

After the slave sends the delay request message, a delay response message is sent by the master. The delay response is sent within a few seconds after the delay request is sent. The delay response contains the master time when the delay request was received by it. We can extract this time by following the steps outlined in section 3.2.

Using the four timestamps, it is possible to calculate the mean propagation delay using the equation for mean propagation delay in section 2.1:

$$t = \frac{(t2 - t1) + (t4 - t3)}{2}$$

Here $t1$ is the master time when the sync message was sent, $t2$ is the slave time when the sync message was received, $t3$ is the slave time when the delay request is sent, and $t4$ is the master time when the delay request is received.

Step 2: Calculating the clock offset

The offset between the master and slave clocks can be calculated using the equation for clock offset in section 2.1.:

$$t2 - t1 - t$$

4 Results

4.1 Clock Offset Calculations Over Time

Fig 4.1 shows the offset between the master clock and slave clock over a time period of roughly 100 seconds. The offset increases with time, which is typical of any two undisciplined clocks. One can see from the plot that the offset increases linearly over the specified time range. This is due to clock drift resulting from a relative tick rate between the master and the slave clock. In fact, the slope of the graph gives the relative tick rate. The relative tick rate between the clocks was found to be 4.93×10^{-5}.

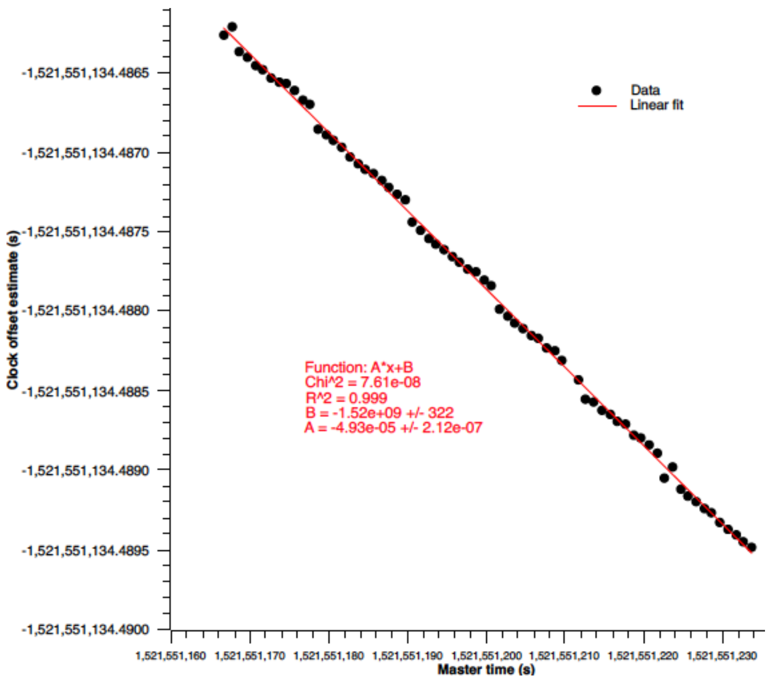

FIGURE 4.1: Plot showing clock offset over time. Data (black dots) fit to a linear model (red line) with parameters shown on the left

4.2 Variation in Time Delay Measurements Between IceCube and ARA Over Different n(z) Models

The index of refraction of ice, n(z), near the firn or below 200 m is known to be changing with the depth of the ice below the surface. Data taken near the South Pole predict an exponential relationship between the depth of the ice and its index

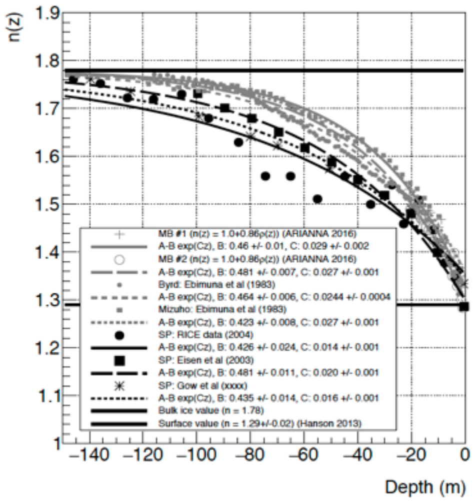

FIGURE 4.2: Plot showing the index of refraction of ice in the firn vs. the depth of the ice at various locations in Antarctica (Plot made by Prof. Jordan Hanson)

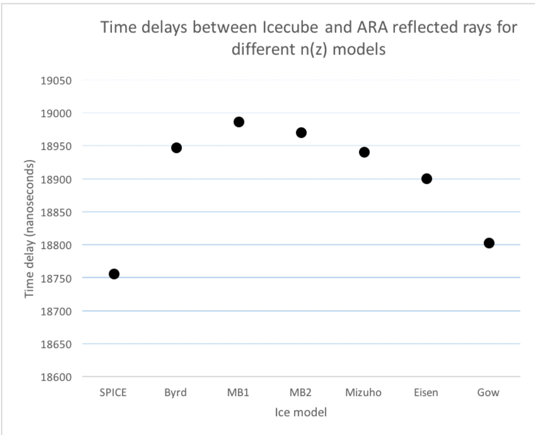

FIGURE 4.3: Plot showing the time delays between ARA and IceCube for reflected rays for different n(z) models

of refraction. However, the fits to the exponential function reveal slightly different parameters depending on the location of the ice. Numerous n(z) models based on the location of the ice in Antarctica have been developed in the past. Fig. 4.2 shows the different n(z) models along with their data.

In the context of doing time coincidence searches between IceCube and ARA, we can compute the time it takes for a signal to be detected by ARA after it is detected at IceCube. This is called the time delay between IceCube and ARA.

Fig. 4.3 and Fig. 4.4 show the simulated time delays between IceCube and ARA for different n(z) models described earlier. The time delays were simulated by using a raytracer and were found for both radio signals reaching the detector after getting reflected from the ice surface above (Fig. 4.3) and radio signals reaching the detector directly without getting reflected (Fig. 4.4). The variation in the time delays between IceCube and ARA was found to be 230.1 ns for reflected rays and 10 ns for direct rays. When looking for reflected rays, we have a broader time window

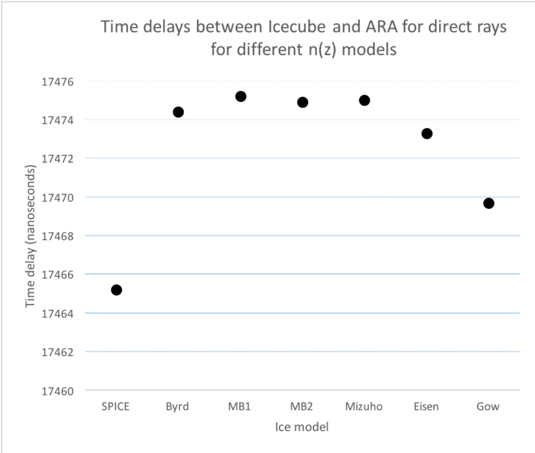

FIGURE 4.4: Plot showing the time delays between ARA and IceCube for direct rays for different n(z) models

and therefore less accurate time synchronization is needed. When looking for direct rays, we have a much narrower time window. Therefore, time synchronization of a higher precision (possibly less than 10 ns) is needed. This is the reason we need precision time protocol to do time synchronization. It can give a precision on the order of nanoseconds.

5 Conclusion

Summary

The implementation of PTP will allow for time synchronization on nanosecond scales. This will lead to more efficient time coincidence searches for neutrino events between the ARA stations and potentially between IceCube and ARA. So far, we have made significant progress towards testing PTP on one of the power distribution boards for ARA. We have been able to reprogram it to exchange PTP messages to and from a local master. We have also implemented the mechanism to gather PTP timestamps from the master and slave clocks. Lastly, we have been able to make initial measurements of the clock offset between the master and slave clocks.

Future work

One particular direction this work could take is trying to discipline the clocks and minimize the clock drift occurring due to the relative tick rate between master and slave clocks. We can also try and test this method with two ARA power distribution boards instead of one power distribution board and a local master. Other challenges to be aware of are clock jitter and noise which need to be filtered using various techniques.

6 References

[1] Correll, Kendall, Nick Barendt, and Michael Branicky. *Design considerations for software only implementations of the IEEE 1588 precision time protocol*, Conference on IEEE. Vol. 1588. 2005.

[2] *Analysis of the second flight of ANtarctic Impulsive Transient Antenna with a focus on filtering techniques.* Dr. Brian Dailey. PhD Dissertation, 2017

[3] A, Connolly and A, Vieregg, *Radio detection of high energy neutrinos*, *ArXiv e-prints* (July 2016), [1607.08232].

[4] P. Allison *et al.*, Astropart. Phys. **35**, 457 (2012) arXiv:1105.2854[astro.ph-IM].

[5] P. Allison *et al.*, Phys. Rev. D **93** (2016) arXiv:1507.08991[astro.ph-IM].

[6] http://www.chronos.co.uk/files/pdfs/cal/TechnicalBrief-IEEE1588v2PTP.pdf